The latex-lab-title package
Changes related to the tagging of the title

LaTeX Project*

v0.85i 2026-10-11

Abstract

1 Introduction

This module contains changes to improve the tagging (in the standard classes) of the title created with the `\maketitle` command. It also improves the setting of the metadata related to the title and the author.

1.1 Tagging of the title

For basic tagging of the printed title there are basically three things to do:

- The actual title should be tagged with a suitable tag.
- The tabular used to format the author list should *not* be tagged as a tabular.
- `\maketitle` redefines footnote internals. These must be made tagging aware.
- There should be some structure that surrounds the whole title block.

1.2 New control over the tagging of title and headings

HTML pages and accessibility API typically do not distinguish titles as distinct from section headings. Titles are marked up as `<h1>`, following heading commands are marked up as `<h2>` etc. This allows the title to be announced by screen readers and take part in navigation.

For PDF 1.7 the PDF association recommends to tag a title like a normal paragraph (https://pdfa.org/how-to-tag-titles-in-pdf-documents/) and to start the tagging of the heading commands with `<h1>`. Such titles are not announced and do not take part in the navigation.

PDF 2.0 introduced a new **Title** structure element which should in principle allow titles to be distinguished, however at the present time it has no defined mapping to the System Accessibility API, and the derivation to HTML maps it to `<div>`. As such, tagging with **Title** is equivalent to tagging with **Div** or **NonStruct** and does not produce accessible titles. Title are simply read as the underlying text.

*Initial implementation done by Ulrike Fischer

Given that the standard mappings to the Accessibility API and to HTML may change, it is not always clear what is the best tagging to use at the present time. The code therefore introduces some options to allow document authors some control here.

LaTeX uses the tag **title-block** to surround the title, author and date. This tag is role mapped to a **Div**.

Author and date are surround by the tags **title-author** and **title-date**. These tags are role mapped to **P**

The title is surrounded by the tag **title-text**.

In PDF 2.0 **title-text** is by default role mapped to **Title**, in PDF 1.7 and in the global Rolemap in PDF 2.0, **Title** it is in turn role mapped to **P**. **part-heading** is mapped to **Title**.

The following options allow to change the tagging. They also affect how headings are tagged.

The options can be set by setting the **title** key either in the preamble with `\tagpdfsetup` or in the `tagging-setup` key of `\DocumentMetadata`.

The key `title` knows the values

Title This is the default already described above: **title-text** is role mapped to **Title**, The tagging of heading commands starts with **H1**. The heading of a `\part` is tagged with **part-heading** and role mapped to **Title**.

H1 **title-text** is role mapped to **H1**. The tagging of heading commands starts with **H2**. **part-heading** is mapped to **H1**. In current systems this produces the most accessible behavior. Documents tagged in this way will pass all mechanically checkable requirements of PDF/UA-2. There is a "human check" that titles should be tagged using **Title** but it seems reasonable to assert that titles in LaTeX documents are structural elements that should be reported to AT and that as currently specified they are not suitable for **Title** which is an anonymous container with no defined accessibility properties.

html In PDF 1.7 this behaves like the `H1` value. In PDF 2.0 **title-text** is role mapped to **h1** in the html namespace, which in turn is role mapped to **Title**. The first heading command is role mapped to **h2** in the HTML namespace, which in turn is role mapped to **H1** in the PDF name space. The global Rolemap is the same as the default case. This mapping can improve derivation to html as derivation can (but doesn't have to) stop when it encounters a tag in the HTML namespace.

aria This additionally adds an Aria-role `heading` with level 1 to the **title-text** structure and Aria-roles `heading` with level 2–7 to the heading commands. **part-heading** gets an Aria-role `heading` with level 1. It doesn't change the role mapping, and can in theory be combined with the other `title` options.

1.3 Metadata

A second task related to title is to store the authors and the title text (or a shorter version) inside the XMP-metadata and (in PDF 1.7 or lower) in the Info dictionary. Currently this can only be set if hyperref is loaded and requires the use of the `pdftitle` and `pdfauthor` keys. The new code therefore extends the `\title` and `\author` commands: They stores their argument and use them at the end of the document for the PDF metadata if the data hasn't been given in another way. The code also gives `\title` and `\author` an optional argument where the PDF title or author can be given with in a key-value syntax. As with hyperref it is possible to store titles in more than one language:

```
\title
  [pdftitle =
     {[en]English Title,[de] Deutscher Titel,[fr]{titre français, avec comma}}]
  {Document title}
```

It is also possible to set a subtitle which is then stored in the XMP-metadata:

```
\title
  [pdfsubtitle =
     {[en]English Subtitle,[de] Deutscher Subtitel,[fr]{subtitre français, avec comma}
  {Document title}
```

If using the `pdfauthor` key authors should be separated by commas, and to hide commas in a name inside braces if needed:

```
\author[pdfauthor = {Bär, Peter Anteater, {Riley, the sloth}}]{\ldots}
```

If hyperref is loaded there is no difference to the `pdftitle` and `pdfauthor` key used in `\hypersetup`. Both can be used (and the last key used will win).

1.4 Open questions and TODOs

- Writing into the Info dictionary needs to convert the input into a PDF string. This is here done with a simple version of hyperref's `\pdfstringdef`, similar code exist also in the generic hyperref driver. This should be moved into a better place module.
- Is it sensible to enhance `\author` and `\title` with an optional argument as done here? An advantage is that it is rather light-weight and doesn't require to decide how this values should be set in `\DocumentMetadata` (and would also work without `\DocumentMetadata`. But a problem could be that various classes and packages already extend this commands with other optional arguments.
- Some of the definitions related to metadata should perhaps be moved into l3pdfmeta.
- Are the names `pdftitle` and `pdfauthor` ok?
- The patch for `\thanks` to get a rlap-footnotemarker looks wrong. This probably means that some configuration option is missing in the footnote code.

2 Implementation

```
1 ⟨*package⟩
2 ⟨@@=tag⟩
3 \ProvidesExplPackage {latex-lab-testphase-title} {\ltlabtitledate} {\ltlabtitleversion}
4   {Changes related to the tagging of the title}
```

2.1 Separating the authors

We redefined `\and` to separate authors in individual structures

```
5 \DeclareRobustCommand\and{%   % \begin{tabular}
```

```
  \end{tabular}\UseTaggingSocket{inline/end}{}%
  \hskip 1em \@plus.17fil%
  \UseTaggingSocket{inline/begin}{tag=title-author,attribute-class=center}%
  \begin{tabular}[t]{c}}%     % \end{tabular}
```

2.2 \maketitle in article class

```
\cs_new_protected:Npn \__tag_patch_thanks:n #1
  {
    \rlap{\footnotemark}
    \protected@xdef\@thanks{\@thanks
      \protect\footnotetext[\the\c@footnote]{#1}}
  }
```

The no-titlepage version of article, report and book

```
\cs_new_protected:Npn \__tag_patch_maketitle:
  {
    \par
    \begingroup
```

Disable table tagging

```
      \tagpdfsetup{table/tagging=false}%
      \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
```

the original definition redefines `\@makefnmark` and `\@makefntext` to get an rlap-mark in the text without affecting the mark in the note (which gives by the way a wrong link area with hyperref). There seem to be currently no good way in the footnote to configure this, so we redefine `\thanks` instead

```
      \cs_set_eq:NN \thanks \__tag_patch_thanks:n
      \if@twocolumn
        \ifnum \col@number=\@ne
          \@maketitle
        \else
          \twocolumn[\@maketitle]%
        \fi
      \else
      \newpage
        \global\@topnum\z@   % Prevents figures from going at top of page.
        \@maketitle
      \fi
      \thispagestyle{plain}\@thanks
    \endgroup
    \setcounter{footnote}{0}%
    \global\let\thanks\relax
    \global\let\maketitle\relax
    \global\let\@maketitle\relax
    \global\let\@thanks\@empty
    \global\let\@author\@empty
    \global\let\@date\@empty
    \global\let\@title\@empty
    \global\let\title\relax
    \global\let\author\relax
    \global\let\date\relax
    \global\let\and\relax
  }
```

We must also change `\@maketitle` to insert a **Title** tag. We have here two options: We can surround the whole block with a **Title** or only the title text (but not both). As default the title text has the **Title** tag and the block is a **Part** (from the symbolic name **para/semantic**), but this can be changed by reassigning the symbolic name **title/block** and changing the plug of the tagging socket to the transparent plug, e.g.,
`\AssignStructureRole{title/block}{Title}`
`\AssignTaggingSocketPlug{title}{transparent}`

Note that in PDF 1.7 the **Title** tag is not a standard tag, it is a user defined and by default rolemapped to **P**. If **Title** should be used for the title block the rolemapping must be adjusted, e.g., to **Part** or **Sect**.

```
\IfSocketExistsF{tagsupport/title}
 {
  \NewTaggingSocket{title}{2}
 }
\NewTaggingSocketPlug{title}{kernel}
 {
   \par
   \group_begin:
     \tagpdfparaOff
     \tag_struct_begin:n{tag=\UseStructureName{title/text},attribute-class=center}
     \tagmcbegin{}#2\tagmcend
     \tag_struct_end:
     \par
   \group_end:
 }
```

An aria attribute to mark up the title-block:

```
\tagpdfsetup{role/new-attribute={aria-banner}{/O/ARIA-1.1/role (banner)}}
\AssignTaggingSocketPlug{title}{kernel}
\cs_new_protected:Npn \__tag_patch_@maketitle:
 {
  \newpage
  \null
  \vskip 2em%
  \begin{center}%
   [tagging-suppress-paras=true,
    tagging-recipe=standalone,
    tag-name=\UseStructureName{title/block},
    tag-attr-class=aria-banner]
  \let \footnote \thanks
```

By default this socket adds **Title** around the title text. In PDF 1.7 **Title** is rolemapped to P and the socket adapts the **para/textblock** to avoid that we get an **P** inside a **P**. One can assign the `transparent` plug to the socket if **Title** should be used for the title block instead.

```
  {\LARGE\UseTaggingSocket{title}{}{\@title}\par}
   \vskip 1.5em%
    {\large
     \AssignStructureRole{para/textblock}{title-author}%
      \lineskip .5em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
```

```
    \vskip 1em%
    {\large
     \AssignStructureRole{para/textblock}{title-date}%
     \@date}%
  \end{center}%
  \par
  \vskip 1.5em
  }

```

The titlepage variant:

```
\cs_new_protected:Npn \__tag_patch_maketitle_page:
  {\begin{titlepage}%
```

disable table tagging

```
  \tagpdfsetup{table/tagging=false}%
  \let\footnotesize\small
  \let\footnoterule\relax
  \let \footnote \thanks
  \null\vfil
  \vskip 60\p@
  \begin{center}%
   [tagging-suppress-paras=true,
    tagging-recipe=standalone,
    tag-name=\UseStructureName{title/block},
    tag-attr-class=aria-banner]
```

By default this socket adds **Title** around the title text. In PDF 1.7 **Title** is rolemapped to P and the socket adapts the **para/textblock** to avoid that we get an **P** inside a **P**. One can assign the `transparent` plug to the socket if **Title** should be used for the title block instead.

```
    {\LARGE\UseTaggingSocket{title}{}{\@title}\par}
    \vskip 3em%
     {\large
      \AssignStructureRole{para/textblock}{title-author}%
      \lineskip .75em%
       \begin{tabular}[t]{c}%
         \@author
       \end{tabular}\par}%
       \vskip 1.5em%
     {\large
      \AssignStructureRole{para/textblock}{title-date}%
      \@date
      \par}%       % Set date in \large size.
  \end{center}\par
  \@thanks
  \vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
```

```
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
  }

```

Map the new commands onto `\maketitle`:

```
\AddToHook{class/article/after}
  {
    \if@titlepage
     \cs_set_eq:NN \maketitle \__tag_patch_maketitle_page:
    \else
     \cs_set_eq:NN \maketitle \__tag_patch_maketitle:
     \cs_set_eq:NN \@maketitle \__tag_patch_@maketitle:
    \fi
  }
\AddToHook{class/report/after}
  {
    \if@titlepage
     \cs_set_eq:NN \maketitle \__tag_patch_maketitle_page:
    \else
    \cs_set_eq:NN \maketitle \__tag_patch_maketitle:
    \cs_set_eq:NN \@maketitle \__tag_patch_@maketitle:
    \fi
  }
\AddToHook{class/book/after}
  {
    \if@titlepage
     \cs_set_eq:NN \maketitle \__tag_patch_maketitle_page:
    \else
    \cs_set_eq:NN \maketitle \__tag_patch_maketitle:
    \cs_set_eq:NN \@maketitle \__tag_patch_@maketitle:
    \fi
  }
```

2.3 Title options

Depending on if there are chapters or not, we have a different offset for the headings from their level.

```
\AddToHook{begindocument}
 {
   \@ifundefined{chapter}
    {\tl_const:Nn \c__tag_title_offset_tl {1}}
    {\tl_const:Nn \c__tag_title_offset_tl {2}}
 }
```

2.3.1 Title (default)

```
\keys_define:nn {__tag/setup}
  {
    title .choice: ,
    title/Title .code:n =
```

```
    {
      \AddToHook{begindocument}
         {
           \@ifundefined{chapter}
            {
              \AssignStructureRole{title/text}{title-text/latex}
              \AssignStructureRole{sec/-1/title}{part-heading/latex}
              \AssignStructureRole{sec/1/title}{section-heading/latex}
              \AssignStructureRole{sec/2/title}{subsection-heading/latex}
              \AssignStructureRole{sec/3/title}{subsubsection-heading/latex}
              \AssignStructureRole{sec/4/title}{paragraph-heading/latex}
              \AssignStructureRole{sec/5/title}{subparagraph-heading/latex}
```

Reassign the default mapping for the global rolemap. TODO: create a simpler interface in tagpdf

```
              \tagpdfsetup
               {
                 role/new-tag={title-text/Title},
                 role/new-tag={part-heading/Title},
                 role/new-tag={section-heading/H1},
                 role/new-tag={subsection-heading/H2},
                 role/new-tag={subsubsection-heading/H3},
                 role/new-tag={paragraph-heading/H4},
                 role/new-tag={subparagraph-heading/H5}
               }
            }
            {
              \AssignStructureRole{title/text}{title-text/latex}
              \AssignStructureRole{sec/-1/title}{part-heading/latex}
              \AssignStructureRole{sec/0/title}{chapter-heading/latex-book}
              \AssignStructureRole{sec/1/title}{section-heading/latex-book}
              \AssignStructureRole{sec/2/title}{subsection-heading/latex-book}
              \AssignStructureRole{sec/3/title}{subsubsection-heading/latex-book}
              \AssignStructureRole{sec/4/title}{paragraph-heading/latex-book}
              \AssignStructureRole{sec/5/title}{subparagraph-heading/latex-book}
              \tagpdfsetup
               {
                 role/new-tag={title-text/Title},
                 role/new-tag={part-heading/Title},
                 role/new-tag={chapter-heading/H1},
                 role/new-tag={section-heading/H2},
                 role/new-tag={subsection-heading/H3},
                 role/new-tag={subsubsection-heading/H4},
                 role/new-tag={paragraph-heading/H5},
                 role/new-tag={subparagraph-heading/H6}
               }
            }
         }
    }
}
```

2.3.2 Aria

The aria option adds aria attributes to **title-text** and the headings. Currently it cannot be disabled again.

```
\keys_define:nn {__tag/setup}
  {
    title/aria .code:n =
      {
```

Set up the attributes and plugs if they do not exist yet.

```
        \IfSocketPlugExistsF {sec/title/begin}{aria}
         {
```

Set up the heading level attributes: `\part` and `\chapter` have always the same attribute. For the other sectioning commands it depends on the class.

```
            \int_step_inline:nn {7}
             {
               \keys_set:nn {__tag/setup}
                {
                  role/new-attribute*=
                    {aria-heading-##1}{/O/ARIA-1.1/role (heading)/aria-level~##1}
                }
             }
```

The plug to add an aria attribute to the title text:

```
          \NewTaggingSocketPlug{title}{aria}
           {
             \par
             \group_begin:
               \tagpdfparaOff
               \tag_struct_begin:n
                {
                 tag=\UseStructureName{title/text},
                 attribute-class={aria-heading-1,center}
                }
               \tagmcbegin{}##2\tagmcend
               \tag_struct_end:%
               \par
             \group_end:
           }
```

We add the attribute class to the display headings:

```
          \NewTaggingSocketPlug{sec/title/begin}{aria}
           {
             \int_compare:nNnTF { \int_eval:n {\use_i:nn ##1} }<{1}
```

`\part` (level -1) and `\chapter` (level 0) should use aria-heading-1 and aria-heading-2

```
              {
                \__tag_sec_title_begin:nnn ##1
                 {attribute-class={aria-heading-\int_eval:n{\use_i:nn ##1 + 2}}}
              }
```

`\section` (level 1) should be mapped to aria-heading 3 in a book class and aria-heading 2 in article.

```
              {
                \__tag_sec_title_begin:nnn ##1
                 {attribute-class={aria-heading-\int_eval:n{\use_i:nn ##1 + \c__tag_title_offs
              }
           }
```

And now the run-in headings. We assume that `\part` and `\chapter` are not a run-in heading. TODO pass the level in a better way

```
          \NewTaggingSocketPlug{sec/title/split}{aria}
           {
            \tag_struct_gput:nnn
              {\tag_get:n{struct_num}}
              {C}
              {/aria-heading-\int_eval:n{ \l__head_level_int +\c__tag_title_offset_tl }}
            \__tag_sec_title_split:
           }
         }
```

And now activate the plugs

```
    \AssignTaggingSocketPlug{title}{aria}
    \AssignTaggingSocketPlug{sec/title/begin}{aria}
    \AssignTaggingSocketPlug{sec/title/split}{aria}
    }
}
```

2.3.3 H1

The `H1` option role maps **title-text** to H1 and headings one level down. We need different versions depending on the PDF version. As tagpdf resets the name space of the heading commands in the begindocument hook, we have to use that too.

```
\pdf_version_compare:NnTF < {2.0}
 {
  \keys_define:nn {__tag/setup}
    {
      title/H1 .code:n =
       {
         \AddToHook{begindocument}
          {
           \@ifundefined{chapter}
           {
             \tagpdfsetup
              {
                role/new-tag={title-text/H1},
                role/new-tag={part-heading/H1},
                role/new-tag={section-heading/H2},
                role/new-tag={subsection-heading/H3},
                role/new-tag={subsubsection-heading/H4},
                role/new-tag={paragraph-heading/H5},
                role/new-tag={subparagraph-heading/H6}
              }
           }
           {
             \tagpdfsetup
              {
                role/new-tag={title-text/H1},
                role/new-tag={part-heading/H1},
                role/new-tag={chapter-heading/H2},
                role/new-tag={section-heading/H3},
                role/new-tag={subsection-heading/H4},
                role/new-tag={subsubsection-heading/H5},
```

```
                  role/new-tag={paragraph-heading/H6},
                  role/new-tag={subparagraph-heading/P}
                }
              }
            }
         }
    }
}
```

PDF 2.0.

```
{
  \keys_define:nn {__tag/setup}
   {
     title/H1 .code:n =
      {
        \AddToHook{begindocument}
         {
           \@ifundefined{chapter}
            {
              \tag_if_tag_known:nnF {title-text}{latex-title-H1}
               {
                \tagpdfsetup
                 {
                   role/new-NS={latex-title-H1}{https://www.latex-project.org/ns/title-H1}{}
                   role/new-tag={tag=title-text,tag-namespace=latex-title-H1,role=H1},
                   role/new-tag={tag=part-heading,tag-namespace=latex-title-H1,role=H1},
                   role/new-tag={tag=section-heading,tag-namespace=latex-title-H1,role=H2},
                   role/new-tag={tag=subsection-heading,tag-namespace=latex-title-H1,role=H3
                   role/new-tag={tag=subsubsection-heading,tag-namespace=latex-title-H1,role
                   role/new-tag={tag=paragraph-heading,tag-namespace=latex-title-H1,role=H5}
                   role/new-tag={tag=subparagraph-heading,tag-namespace=latex-title-H1,role=
                 }
               }
              \AssignStructureRole{title/text}{title-text/latex-title-H1}
              \AssignStructureRole{sec/-1/title}{part-heading/latex-title-H1}
              \AssignStructureRole{sec/1/title}{section-heading/latex-title-H1}
              \AssignStructureRole{sec/2/title}{subsection-heading/latex-title-H1}
              \AssignStructureRole{sec/3/title}{subsubsection-heading/latex-title-H1}
              \AssignStructureRole{sec/4/title}{paragraph-heading/latex-title-H1}
              \AssignStructureRole{sec/5/title}{subparagraph-heading/latex-title-H1}
             }
```

for book classes

```
            {
              \tag_if_tag_known:nnF {title-text}{latex-title-H1-book}
                {
                  \tagpdfsetup
                   {
                     role/new-NS={latex-title-H1-book}{https://www.latex-project.org/ns/title
                     role/new-tag={tag=title-text,tag-namespace=latex-title-H1-book,role=H1},
                     role/new-tag={tag=part-heading,tag-namespace=latex-title-H1-book,role=H1
                     role/new-tag={tag=chapter-heading,tag-namespace=latex-title-H1-book,role
                     role/new-tag={tag=section-heading,tag-namespace=latex-title-H1-book,role
                     role/new-tag={tag=subsection-heading,tag-namespace=latex-title-H1-book,r
                     role/new-tag={tag=subsubsection-heading,tag-namespace=latex-title-H1-boo
```

```
                    role/new-tag={tag=paragraph-heading,tag-namespace=latex-title-H1-book,ro
                    role/new-tag={tag=subparagraph-heading,tag-namespace=latex-title-H1-book
                  }
               }
             \AssignStructureRole{title/text}{title-text/latex-title-H1-book}
             \AssignStructureRole{sec/-1/title}{part-heading/latex-title-H1-book}
             \AssignStructureRole{sec/0/title}{chapter-heading/latex-title-H1-book}
             \AssignStructureRole{sec/1/title}{section-heading/latex-title-H1-book}
             \AssignStructureRole{sec/2/title}{subsection-heading/latex-title-H1-book}
             \AssignStructureRole{sec/3/title}{subsubsection-heading/latex-title-H1-book}
             \AssignStructureRole{sec/4/title}{paragraph-heading/latex-title-H1-book}
             \AssignStructureRole{sec/5/title}{subparagraph-heading/latex-title-H1-book}
           }
         }
       }
   }
}
```

2.3.4 html

In PDF 1.7 this behaves like the `H1` value. In PDF 2.0 **title-text** is role mapped to **h1** in the html namespace, which in turn is role mapped to **Title**. The first heading command is role mapped to **h2** in the HTML namespace, which in turn is role mapped to **H1** in the PDF name space. The global Rolemap is the same as the default case. This mapping can improve derivation to html as derivation can (but doesn't have to) stop when it encounters a tag in the HTML namespace.

```
\pdf_version_compare:NnTF < {2.0}
 {
  \keys_define:nn {__tag/setup}
    {
      title/html .code:n =
       {
        \keys_set:nn {__tag/setup}{title=H1}
       }
    }
}
{
  \keys_define:nn {__tag/setup}
    {
      title/html .code:n =
       {
        \tag_if_tag_known:nnF { h1 }{ html }
         {
          \keys_set:nn {__tag/setup}
            {
             role/new-NS={html}{http://www.w3.org/1999/xhtml}{},
             role/new-tag={tag=h1,tag-namespace=html,role=Title},
             role/new-tag={tag=h2,tag-namespace=html,role=H1},
             role/new-tag={tag=h3,tag-namespace=html,role=H2},
             role/new-tag={tag=h4,tag-namespace=html,role=H3},
             role/new-tag={tag=h5,tag-namespace=html,role=H4},
             role/new-tag={tag=h6,tag-namespace=html,role=H5}
            }
```

```
          }
         \AddToHook{begindocument}
          {
           \@ifundefined{chapter}
            {
             \AssignStructureRole{title/text}{h1}
             \AssignStructureRole{sec/-1/title}{h1}
             \AssignStructureRole{sec/1/title}{h2} %section role mapped to H1
             \AssignStructureRole{sec/2/title}{h3}
             \AssignStructureRole{sec/3/title}{h4}
             \AssignStructureRole{sec/4/title}{h5}
             \AssignStructureRole{sec/5/title}{h6}
            }
            {
             \AssignStructureRole{title/text}{h1}
             \AssignStructureRole{sec/-1/title}{h1}
             \AssignStructureRole{sec/0/title}{h2}
             \AssignStructureRole{sec/1/title}{h3} %section role mapped to H1
             \AssignStructureRole{sec/2/title}{h4}
             \AssignStructureRole{sec/3/title}{h5}
             \AssignStructureRole{sec/4/title}{h6}
            }
          }
        }
    }
}
```

2.4 Helper commands to set metadata

Some temp variables

```
\str_new:N \g__tag_title_tmpa_str
\str_new:N \l__tag_title_tmpa_str
\tl_new:N  \l__tag_title_tmpa_tl
\seq_new:N \l__tag_title_tmpa_seq
```

We ensure that the default definitions of `\@title` and `\@author` are robust:

```
\protected\def\@title{\@latex@error{No~\noexpand\title given}\@ehc}
\protected\def\@author{\@latex@warning@no@line{No~\noexpand\author given}}
```

A helper command to convert the title into a pdfstring similar to `\pdfstringdef`.

```
\cs_new_protected:Npn \__tag_title_pdfstring:nnN #1 #2 #3 % #1 text, #2 e.g. utf16/hex
  {
    \group_begin:
    \pdf_purify:nN {#1}\l__tag_title_tmpa_tl
    \pdf_string_from_unicode:nVN { #2 } \l__tag_title_tmpa_tl \l__tag_title_tmpa_str
    \str_gset_eq:NN \g__tag_title_tmpa_str\l__tag_title_tmpa_str
    \group_end:
    \str_set_eq:NN #3 \g__tag_title_tmpa_str
  }
\cs_generate_variant:Nn\__tag_title_pdfstring:nnN {e}
```

2.5 Extend title to set metadata

At first a variable to store the title, as `\@title` is emptied by LaTeX.

```
\tl_new:N \g__tag_title_title_tl
```

Now we redefine `\title` so that it stores the title, and processes keys in the optional argument. We use `hyp` as module name for the key as this means that if hyperref is loaded its definition of `pdftitle` will be used – at some time probably this should be moved out of hyperref so that we have only one definition.

```
\RenewDocumentCommand\title{O{}m}
 {
    \gdef\@title{#2}
    \tl_gset_eq:NN\g__tag_title_title_tl\@title
    \keys_set:nn {hyp}{#1}
 }
```

Now we define the `pdftitle` key. This is more or less the same definition as in the generic hyperref driver.

```
\regex_new:N\l__tag_title_optlang_regex
\regex_set:Nn\l__tag_title_optlang_regex {\A\[([A-Za-z\-]+)\](.*)}
\cs_generate_variant:Nn \clist_item:nn {on}
```

and now the keys.

```
\keys_define:nn { hyp }
   {
     pdftitle  .code:n =
       {
         \tl_if_blank:nTF {#1}
           {
             \pdfmanagement_remove:nn {Info}{Title}
           }
           {
             \tl_set:Ne\l__tag_title_tmpa_tl {\clist_item:on{#1}{1}}
             \regex_extract_once:NVN
                \l__tag_title_optlang_regex
                \l__tag_title_tmpa_tl
                \l__tag_title_tmpa_seq
             \seq_if_empty:NTF\l__tag_title_tmpa_seq
              {
                \__tag_title_pdfstring:nnN {#1}{utf16/hex}\l__tag_title_tmpa_str
              }
              {
                \__tag_title_pdfstring:enN
                   {\seq_item:Nn \l__tag_title_tmpa_seq{3}}{utf16/hex}\l__tag_title_tmpa_str
              }
             \str_if_eq:VnF\l__tag_title_tmpa_str{<FEFF>}
               {
                 \pdfmanagement_add:nne {Info}{Title}{\l__tag_title_tmpa_str}
               }
           }
          \AddToDocumentProperties[hyperref]{pdftitle}{#1}
       }
    ,pdfsubtitle .code:n = { \AddToDocumentProperties[hyperref]{pdfsubtitle}{#1} }
    }
```

2.6 Extend \author to set metadata

At first a variable to store the authors, as `\@author` is emptied by LaTeX.

```
\tl_new:N \g__tag_title_author_tl
```

Now we redefine `\author` so that it stores the authors, and processes keys in the optional argument. We use `hyp` as module name for the key as this means that if hyperref is loaded its definition of `pdfauthor` will be used – at some time probably this should be moved out of hyperref so that we have only one definition.

```
\RenewDocumentCommand\author{O{}m}
 {
    \gdef\@author{#2}
    \tl_gset_eq:NN\g__tag_title_author_tl\@author
    \keys_set:nn {hyp}{#1}
 }
```

Now we define the `pdfauthor` key. This is more or less the same definition as in the generic hyperref driver.

```
\keys_define:nn { hyp }
   {
     pdfauthor  .code:n =
       {
         \tl_if_blank:nTF {#1}
           {
             \pdfmanagement_remove:nn {Info}{Author}
           }
           {
             \tl_set:Ne\l__tag_title_tmpa_tl {\clist_item:on{#1}{1}}
             \regex_extract_once:NVN
                \l__tag_title_optlang_regex
                \l__tag_title_tmpa_tl
                \l__tag_title_tmpa_seq
             \seq_if_empty:NTF\l__tag_title_tmpa_seq
              {
                \__tag_title_pdfstring:nnN {#1}{utf16/hex}\l__tag_title_tmpa_str
              }
              {
                \__tag_title_pdfstring:enN
                   {\seq_item:Nn \l__tag_title_tmpa_seq{3}}{utf16/hex}\l__tag_title_tmpa_str
              }
             \str_if_eq:VnF\l__tag_title_tmpa_str{<FEFF>}
               {
                 \pdfmanagement_add:nne {Info}{Author}{\l__tag_title_tmpa_str}
               }
           }
          \AddToDocumentProperties[hyperref]{pdfauthor}{#1}
       }
   }
```

2.7 Fallback for classes and packages that redefine \title or \author

If a class redefines `\author` and `\title` again, we try to retrieve at least the values.

```
\AddToHook{cmd/maketitle/before}
 {
  \tl_gset_eq:NN \g__tag_title_author_tl\@author
  \tl_gset_eq:NN \g__tag_title_title_tl\@title
 }
```

2.8 Finalize document

At last we set the title and the author at the end of document if that hasn't happened yet:

```
\AddToHook{shipout/lastpage}
 {
   \tl_if_empty:eT{\GetDocumentProperty{hyperref/pdftitle}}
      {
        \group_begin:
        \cs_set_eq:NN\thanks \use_none:n
        \pdf_purify:nN { \g__tag_title_title_tl }\l__tag_title_tmpa_tl
        \tl_if_blank:oF { \l__tag_title_tmpa_tl }
         { \keys_set:ne{hyp}{pdftitle={{\exp_not:V\l__tag_title_tmpa_tl}}} }
        \group_end:
      }
   \tl_if_empty:eT{\GetDocumentProperty{hyperref/pdfauthor}}
      {
        \group_begin:
        \cs_set_eq:NN\thanks \use_none:n
        \cs_set:Npn \and {,}
        \pdf_purify:nN { \g__tag_title_author_tl }\l__tag_title_tmpa_tl
        \tl_if_blank:oF { \l__tag_title_tmpa_tl }
          { \keys_set:ne{hyp}{pdfauthor={\exp_not:V\l__tag_title_tmpa_tl}} }
        \group_end:
      }
```

force display title, if an UA-standard is detected.

```
   \tl_if_empty:eF{\GetDocumentProperty{document/pdfstandard-UA}}
     {
       \pdfmanagement_add:nnn {Catalog / ViewerPreferences } { DisplayDocTitle } { true }
     }
 }
\DeclareHookRule{shipout/lastpage}{latex-lab-testphase-title}{before}{pdfmanagement-testphase
⟨/package⟩
⟨*latex-lab⟩
\ProvidesFile{title-latex-lab-testphase.ltx}
        [\ltlabtitledate\space v\ltlabtitleversion\space
         Changes related to the tagging of the title]

\RequirePackage{latex-lab-testphase-title}

⟨/latex-lab⟩
```